LTPDA

a MATLAB toolbox for accountable and reproducible data analysis

M Hewitson for the LTP team

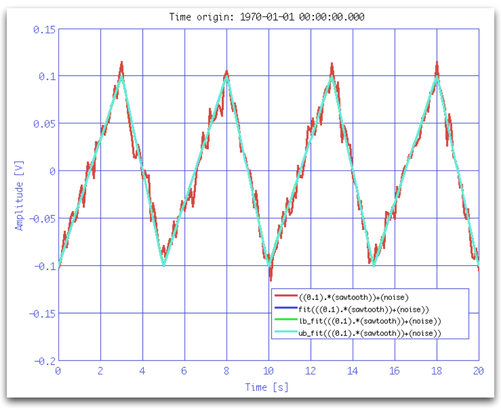
Introduction

LTPDA is a MATLAB toolbox that offers an object-oriented approach to data analysis. The user creates and modifies different types of user objects in order to build up a signal processing pipeline. Each user object keeps a full record of all processing steps it goes through. Because of this, the full processing history of each object can be viewed at any time, and used to reproduce the object, or the processing pipeline that created it.

Implementation

The toolbox is implemented as a set of classes programmed in standard MATLAB. There are a number of user classes which represent objects intended to be manipulated by the user. The methods of these user classes provide the user the ability to do:

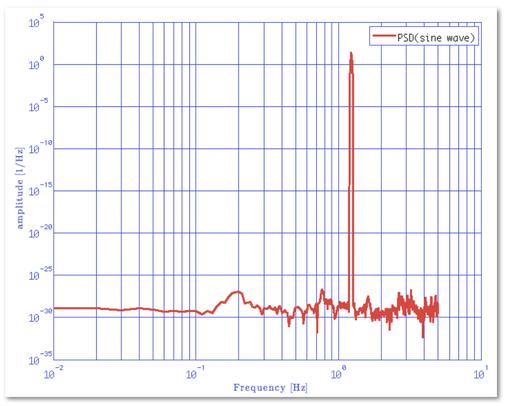
- Signal-conditioning
- Spectral-analysis
- Curve-fitting and parameter estimation
- Pole/zero modelling
- State-space modelling
- Dimensional analysis
- Digital filtering
- ... and much more.



Due to the object-oriented design of the toolbox, script writing is vastly simplified and typically only requires the application of class methods to the objects created by the user. For example, the following code extract creates a synthetic time-series signal and then computes and plots an estimate of the power spectral density:

```
a = ao(plist('waveform', 'sine wave', 'fs', 10, 'nsecs', 100));
a.psd
a.iplot
```

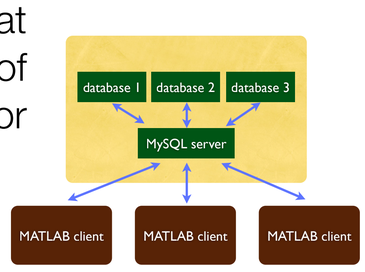
Running these commands yields the plot shown here:



As well as the user objects, there are many other object classes implemented which provide the infrastructure used by the user classes.

Centralised Data Access

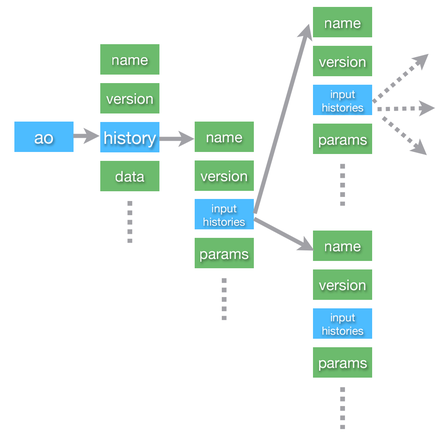
In order to provide concurrent storage and retrieval of LTPDA user objects, the toolbox has a built in client that can talk to an LTPDA server. The server is a database system that stores the LTPDA objects and a wealth of meta-data which can be used to search for specific objects.



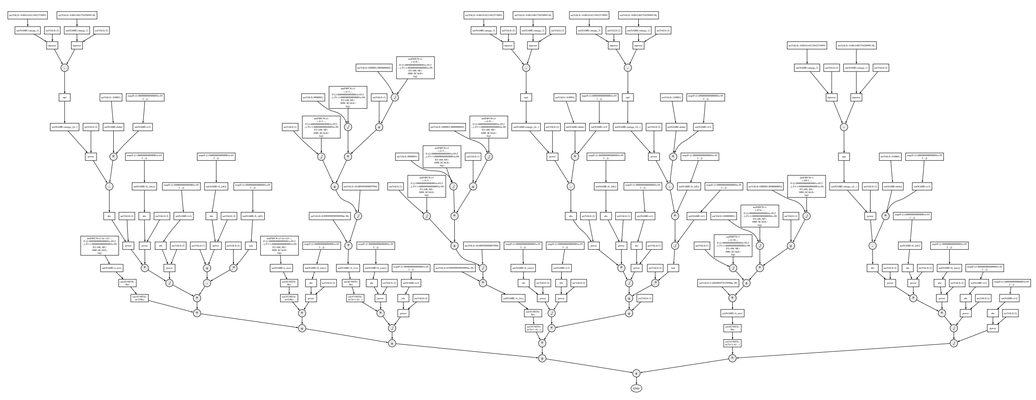
Tracking History

One of the main features of the toolbox is the ability to track the full processing history of each user object. At every step of the analysis a record of the algorithm and the parameters used is attached to the processing history of each input object. In this way, a full history tree is built up. This can be used to, for example,

- view the history of the object
- recreate the object
- produce a script/pipeline based on this history



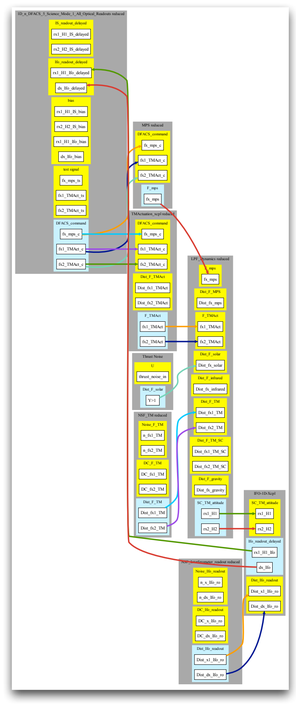
History trees can be viewed either directly in MATLAB, or using the well-known graphing application, Graphviz (http://www.graphviz.org/):



User Objects

The user objects form the core of the toolbox's capabilities. The following classes of user object are implemented:

class	description
ao	Analysis objects containing time-series, frequency-series, x-y data series, vectors or matrices
plist	Parameter lists for configuring algorithms.
ssm	State-space model objects
miir	IIR digital filter representation
mfir	FIR digital filter representation
pzmodel	Transfer function representation as S-domain poles and zeros
rational	Transfer function represented as a rational function of S
parfrac	Partial fraction representation of transfer functions
timespan	Representation of an interval of time



Each user object can be saved and loaded from disk in an XML format, and submitted and retrieved to/from and LTPDA repository.